

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: INVERSE HALFTONING PROCESS

APPLICANT: WILLIAM W. MACY, JR.

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL298427975US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

May 14, 2001

Date of Deposit

Signature

Michael E. Augustine, Sr.

Typed or Printed Name of Person Signing Certificate

INVERSE HALFTONING PROCESS

TECHNICAL FIELD

This invention relates to inverse halftoning.

5

BACKGROUND

Digital halftoning recreates an original shaded image using light and dark pixels. Varying the density and placement of the light and dark pixels in the resulting
10 digitally-halftoned image approximates shading gradations in the original image. Inverse halftoning is a process of enhancing a digitally-halftoned image so that it more closely resembles the original shaded image.

15

DESCRIPTION OF THE DRAWINGS

Fig. 1 is a flowchart showing an inverse-halftoning process according to one embodiment of the invention.

Fig. 2 is a view of pixels in images processed by the inverse halftoning process.

20

Fig. 3 is a block diagram of computer hardware on which the inverse halftoning process may be implemented.

DESCRIPTION

Referring to Fig. 1, a process 5 is shown for performing inverse halftoning on a halftoned image. Process 5 is basically a two-part process, with enhancements. In Fig. 1, the enhancements are shown by dotted lines.

In the basic two-part process, process 5 lightly smoothes (101) the halftoned image and then performs (103) lowpass filtering on the halftoned image. To lightly smooth the halftoned image, process 5 applies a two-dimensional (e.g., $N \times N$, where $N > 1$) filter to each pixel in the image.

In one embodiment, a strongly-peaked 3×3 filter is used to perform the light smoothing. Examples of filters to produce different shading effects are (A) and (B) below.

	$\begin{bmatrix} 1 & 6 & 1 \\ 6 & 8 & 6 \\ 1 & 6 & 1 \end{bmatrix} * (1/36)$		$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 1 \end{bmatrix} * (1/20)$
(A)		(B)	

A filter is applied to an image as follows. The filter matrix is overlaid on a block of $N \times N$ (e.g., 3×3) pixels. Each numerical value in the filter matrix is multiplied by a corresponding underlying pixel value. The resulting products are added together and multiplied by the reciprocal of the sum

of the filter values (e.g., $(1/36)$ and $(1/20)$ above). The resulting value is stored as a pixel in an intermediary smoothed image. The pixel in the intermediary smoothed image is at the same location in the smoothed image as the center
5 pixel in the block of $N \times N$ pixels that were filtered in the original image.

To process other pixels, the filter matrix is moved to an adjacent pixel in the original image. The filter matrix may be moved horizontally or vertically. Referring to Fig. 2, as
10 each pixel 6 in an original image 7 is processed, the resulting filtered pixel is stored at a corresponding location 8 in a smoothed image 9. The resulting smoothed image, or portion thereof, is lowpass filtered to produce an enhanced image. Lowpass filtering is performed in order to reduce
15 artifacts in the image. Examples of artifacts include unwanted spots or image aberrations.

Process 5 performs (103) lowpass filtering using a one-dimensional lowpass filter having a relatively sharp high-frequency cutoff. Examples of well-known lowpass filters that
20 have sharp high-frequency cutoffs, and that may be used, include Hamming and Kaiser filters. In this embodiment, a one-dimensional filter with N ($N \geq 1$) taps is used; however, a two-dimensional filter, whose dimensions in some, but not all,

instances will be NxN, may be used to perform the lowpass filtering. The lowpass filter may have any number of "taps", e.g., 11, 7, 3, etc., and may be weighted by a sinc function to vary filter width. The lowpass filter is moved over the smoothed image, and filters the smoothed image, in much the same way as the 3x3 filter described above.

However, unlike the case of filtering with a 3x3 two-dimensional matrix, filtering with a one-dimensional matrix is done by making two passes, one horizontal and one vertical, over the image. First, the lowpass matrix is overlaid along either a block of row pixels and moved horizontally or a block of column pixels and moved vertically to create an intermediate image. On the second pass, the filter is applied to the intermediate image along the direction not applied in the first pass. For example, a filter having the matrix values

$$\begin{matrix} 1 & 6 & 1 \end{matrix}$$

may be used along a row. As above, each numerical value in the filter is multiplied by a corresponding underlying pixel value. The resulting products are added together and multiplied by the reciprocal of the sum of the filter values. The resulting value is assigned to a pixel in an intermediate

5

10

15

20

To process other pixels in the smoothed image, the lowpass filter is moved to an adjacent pixel in the smoothed image. That adjacent pixel is then processed using the two-pass filtering described above. As each pixel in the smoothed
5 image is processed, the resulting filtered pixel value is stored at a corresponding location in the enhanced image. By repeating this process for plural pixels in the smoothed image, process 5 generates (104) an enhanced, inverse-halftoned, version of the smoothed image.

10 Additional features may be incorporated into process 5. These features are shown in dotted lines on Fig. 1. For example, process 5 may detect (102) edges in the smoothed image and ignore the edges during lowpass filtering (103), i.e., only lowpass-filter the non-edge areas of the smoothed
15 image. Thus, in this case, the enhanced image is comprised of the edge areas of the smoothed image and lowpass-filtered non-edge areas of the smoothed image. One reason for ignoring the edges during lowpass filtering is that lowpass filtering tends to blur the image. To ensure that the edges remain relatively
20 defined, process 5 may detect those edges and perform lowpass filtering selectively, such that the edges are not lowpass filtered.

To detect edges in the smoothed image, process 5 applies an edge filter to the smoothed image. The edge filter may be an NxN filter, which processes pixels in substantially the same manner as the NxN filter described above. That is, the filter matrix is overlaid on a block of NxN (e.g., 3x3) pixels. Each numerical value in the filter is multiplied by a corresponding underlying pixel value. The resulting products are added together to obtain a value. In this case, unlike above, the value is not multiplied by the reciprocal of the sum of the pixel values. If the resulting value has a large magnitude, that is an indication of an edge in the image.

One example of an edge detection filter matrix that may be used in this embodiment is as follows:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Applying this edge detection filter to a flat surface of a pixel would result in a near zero value, since the pixels underneath the first column (-1's) would cancel the pixels underneath the third column (1's). If an edge is present at the center column, the value resulting from applying the filter over the center pixel and its adjacent eight pixels

would result in a non-zero magnitude because values along one center are different from other adjacent values. The higher this magnitude, the more pronounced the edge.

5 Process 5 compares the value that results from applying the edge detection filter to a threshold value. Edges whose values exceed the threshold are ignored during lowpass filtering. This threshold value is set beforehand and is set, as desired, to ignore some edges and account for others. The higher that the threshold value is set, the more edges the process 5 ignores during lowpass filtering (103). That is, if 10 the threshold is set relatively high, only highly-pronounced edges will not be subjected to lowpass filtering, whereas if the threshold is set relatively low, relatively less-pronounced edges will also not be subjected to lowpass 15 filtering.

Process 5 may also apply (105) a median filter to the resulting enhanced image. The median filter is designed to reduce artifacts, such as spots or other aberrations, in the enhanced image. The median filter may also be an NxN filter, 20 which is applied in the manner described above, to further smooth the enhanced image. The median filter may be applied to non-edge portions of the enhanced image only, if desired. In this case, an edge detection filter of the type described

above is used to detect edges in the enhanced image prior to applying the median filter to the enhanced image.

Fig. 3 shows a computer 28 on which process 5 may be implemented. Computer 28 includes a processor 30, a memory 32, and a storage medium 34 (see view 36). Storage medium 34 stores data 38 for one or more images and machine-executable instructions 40 that are executed by processor 30 out of memory 32 to perform inverse halftoning process 5.

Although a personal computer is shown in Fig. 3, process 5 is not limited to use with the hardware and software of Fig. 3. It may find applicability in any computing or processing environment. Process 5 may be implemented in hardware, software, or a combination of the two. Process 5 may be implemented in computer programs executing on programmable computers or other machines that each include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage components), at least one input device, and one or more output devices. Program code may be applied to data entered using an input device (e.g., a mouse or keyboard) to perform process 5 and to generate output information.

Each such program may be implemented in a high level procedural or object-oriented programming language to

communicate with a computer system. However, the programs can be implemented in assembly or machine language. The language may be a compiled or an interpreted language.

Each computer program may be stored on a storage medium/
5 article (e.g., CD-ROM, hard disk, or magnetic diskette) that is readable by a general or special purpose programmable computer for configuring and operating the computer when the storage medium or device is read by the computer to perform process 5. Process 5 may also be implemented as a machine-
10 readable storage medium, configured with a computer program, where, upon execution, instructions in the computer program cause a machine to operate in accordance with process 5.

The invention is not limited to the specific embodiments described above. For example, a Sobel gradient operator may
15 be used for edge detection instead of, or in addition to, the NxN matrix described above. The invention is not limited to the specific filters described herein, to their numerical values, or to their dimensions. The invention can be used to perform inverse halftoning on images received over a network
20 or on any other types of halftoned images.

Other embodiments not described herein are also within the scope of the following claims.

What is claimed is: